

REPORT

그림판 만들기

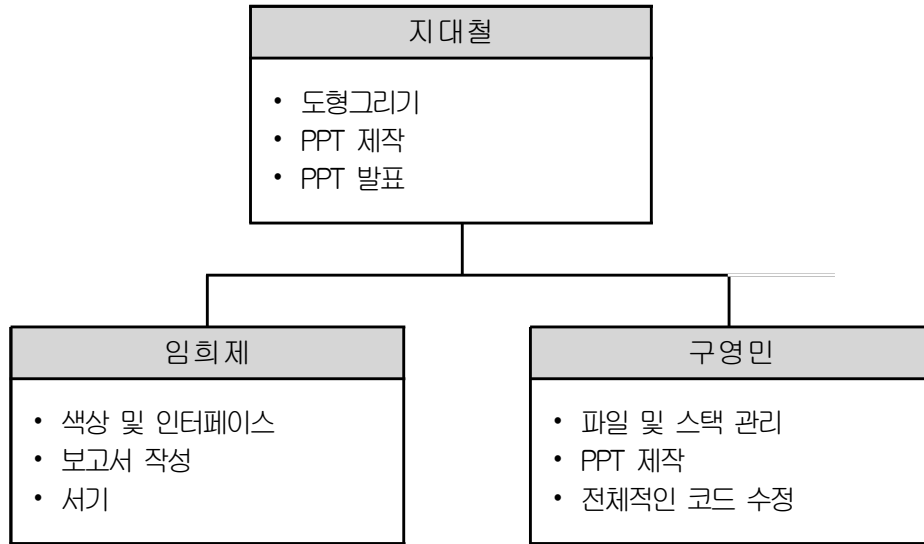


과목명	자바와 객체지향프로그래밍	
교수명	신 석 훈 교수님	
학과	소프트웨어학과	
학번/이름	2012122284	지대철
	2015125068	임희제
	2016125003	구영민
제출일	2016년 12월 12일	

1) 팀 구성 및 업무분담

조장

팀원



2) 계획 및 일정

활 동 내 용		진행 과정						
		28일	29일	30일	1일	2일	3일	4일
○ Project planning - 팀 구성(3인) 및 회의 - 목표, 기능, 성능, 자료조사, 추진일정 ○ 설계 - Editor의 개념도 - CLASS DIAGRAM - GUI Interface Design - Class Interface Design - 기타 구현단계 필요한 상세정보정리 ○ 구현 - 프로그램 작성 - 프로그램 검증 - 프로그램 시현								
목표 대비 진행 정도	기간별(%)	25	25	25	25			
	누적(%)	25	50	75	100			

3) 문제 분석 및 설계

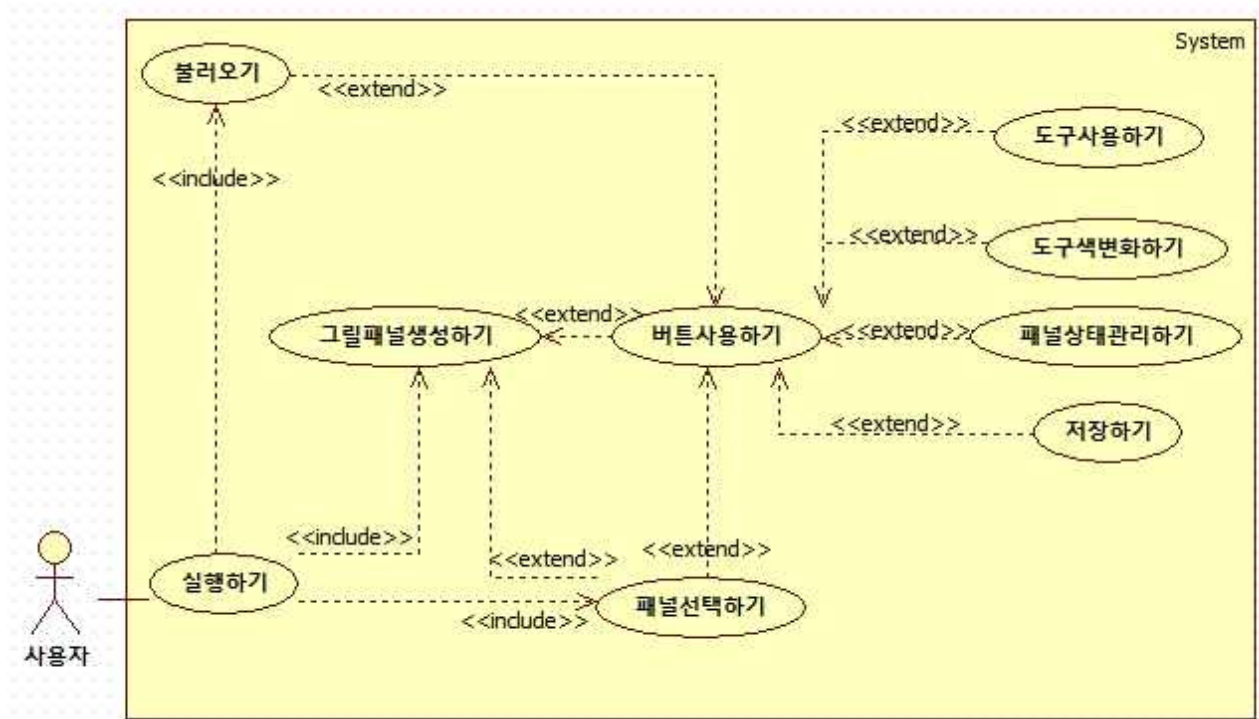
❖기본요구사항

- 저장/불러오기
- 자유드로잉
- 직선/사각형/원 드로잉
- 색상 채우기
- 지우기
- 전경/배경 색상선택

❖추가요구사항

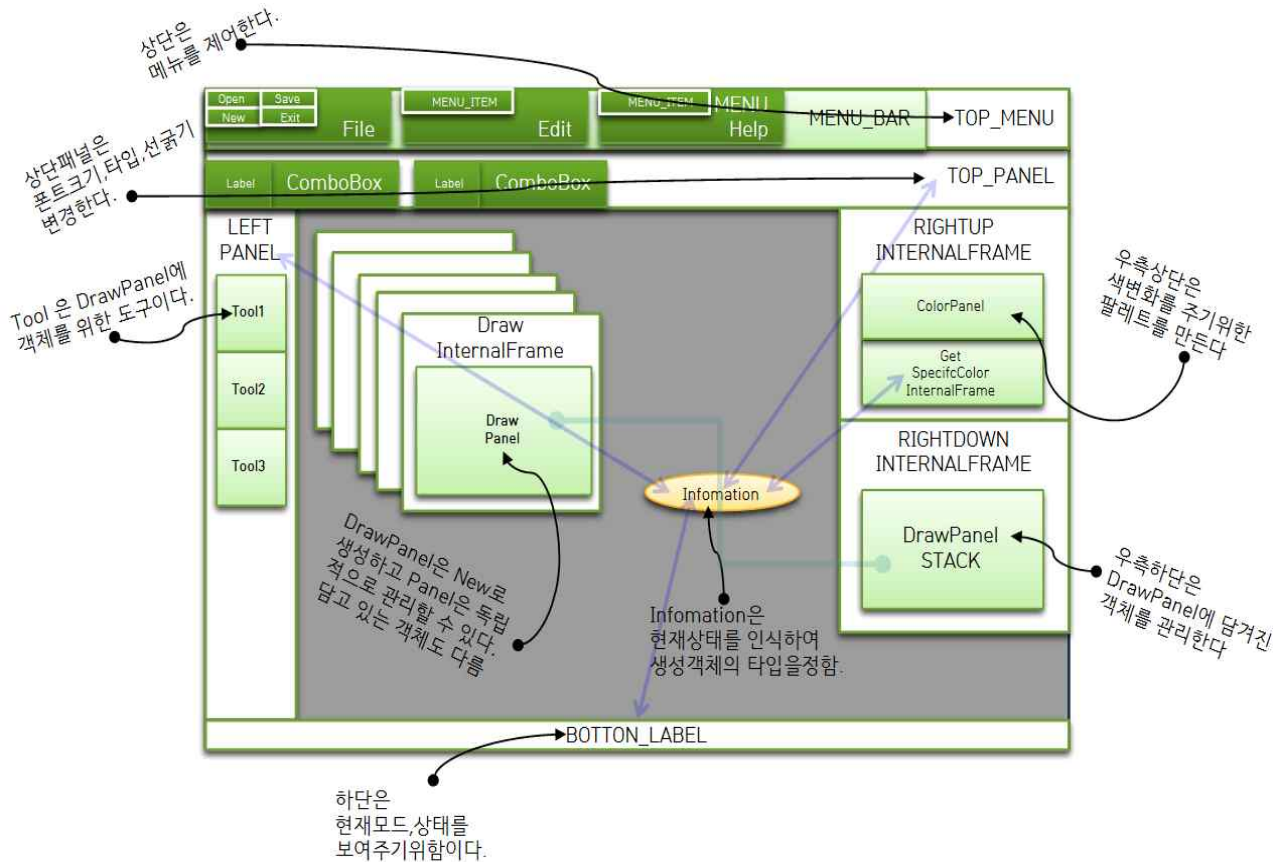
- 드로잉된 개체선택기능
- 다각형드로잉
- 직선점드로잉
- 그라데이션 채우기
- 투명도
- 레이어
- JPEG/PNG 등 다른 형식으로 저장/불러오기
- 기타 추가기능구현

❖Use-Case diagram 설계



사용자가 프로그램을 실행하였을 때, 할 수 있는 행동들을 Use-Case로 설계하였다. 사용자가 실행을 한 이후로 그리 패널을 생성하기, 패널 선택하기, 불러오기를 할 수 있으며, 해당 기능을 사용한 후에는 도구사용하기, 도구색변화하기, 패널 상태 관리하기, 저장하기로 버튼을 사용할 수 있다.

❖GUI 인터페이스 설계



총 3개의 패널과 3개의 프레임 1개의 메뉴, 라벨을 갖는다. 메뉴는 상단에 부착되어 위치의 이동이 없다. 이하 3개의 패널은 상단패널(Top panel), 좌측패널(Left panel), 그림패널 (Draw panel)이다. 상단패널은 폰트, 크기를 조절하고, 좌측패널은 도구들을 나열한다.

움직일 수 있는 프레임으로 구성된 세 개의 프레임은 그림패널을 넣고 있는 그림 프레임 (DrawInternalFrame), 우측상단 프레임 (RightUp InternalFrame), 우측하단 프레임 (RightDown InternalFrame) 이다. 그림 프레임은 그림 패널을 가지고 있으며, 이 패널은 마우스로 들어온 값을 받아 객체를 생성하고 정보를 관리한다. 이후 패널의 독립성으로 여러 패널들은 각자 가진 고유의 값이 있음을 우측하단 프레임에 구현 할 스택구조를 볼 수 있다. 우측 상단의 프레임은 팔레트로 도구를 사용할 때 색을 변경할 수 있다. 추가적으로 더 많은 색을 이용하기 위해 하나의 프레임을 열 것이다. 우측 하단의 프레임은 그림 패널에서 그려진 객체들을 보여주며, 스택내부를 볼 수 있고 기능으로 객체 지정하여 지우거나, 이전상태로 되돌리기 이전상태로 되돌아간 경우 이후상태로 되돌리는 것과 스택 전체를 비우는 기능을 할 수 있다.

메뉴로는 저장하기, 불러오기, 새로운 그림프레임 열기, 종료하기로 구성한다. Information으로 노란색 타원은 내부에 존재하지만 겉으로 보이지는 않기에 사각형이 아닌 원으로 표현하였다. 이는 정보를 가지고 있으며 GUI 상에 사용자가 하는 기능을 기억한다.

4) 문제 해결

❖파일 쓰기/읽기

- 저장/불러오기, JPEG/PNG 등 다른 형식으로 저장/불러오기

```
// JMenuItem 으로 열기 기능을 구현하고 액션 리스너를 추가함
open = new JMenuItem("Open");
open.setMnemonic('O');
open.addActionListener(e -> {
    // 파일 선택기 객체
    JFileChooser fileChooser = new JFileChooser();

    // 파일 필터 만들기
    FileNameExtensionFilter filter = new FileNameExtensionFilter("JPG, PNG", "jpg", "png");
    fileChooser.setFileFilter(filter);

    fileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
    int judge = fileChooser.showOpenDialog(this);

    switch (judge) {
        case JFileChooser.APPROVE_OPTION:
            File x = fileChooser.getSelectedFile();
            try {
                // 파일을 읽는다
                BufferedImage image = ImageIO.read(x);

                // 가져온 파일을 DrawFrame 에 집어넣는다.
                MainFrame.getInstance().addDrawFrame(x.getPath());
                Picture picture = new Picture(0, 0, getWidth(), getHeight());
                picture.setGraphics(image); // 이미지를 주입한다.
                MainDesktopPane.getInstance().getDrawFrame(x.getPath()).draw.figureSet.addElement(picture);
            } catch (Exception e1) {
                e1.printStackTrace();
            }
            break;
    }
});

save = new JMenuItem("Save");
save.setMnemonic('S');
save.addActionListener(e -> {
    File x;
    JFileChooser jfilechooser = new JFileChooser();

    // 확장자를 선택 목록에 추가한다.
    jfilechooser.setFileFilter(new FileNameExtensionFilter("PNG", "png"));
    jfilechooser.addChoosableFileFilter(new FileNameExtensionFilter("JPG", "jpg"));

    jfilechooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
    int judge = jfilechooser.showSaveDialog(null);

    switch (judge) {
        case JFileChooser.APPROVE_OPTION:
            x = jfilechooser.getSelectedFile();
            // 확장자가 붙어 있지 않으면 확장자를 붙인다.
```

```

if (jfilechooser.getFileFilter().getDescription().equals("JPG") && !x.getName().endsWith("jpg")) {
    x = new File(x.getPath() + ".JPG");
}
if (jfilechooser.getFileFilter().getDescription().equals("PNG") && !x.getName().endsWith("png")) {
    x = new File(x.getPath() + ".PNG");
}
System.out.println(jfilechooser.getFileFilter().getDescription());
try {
    // 활성화된 JPanel 을 가져와 캡처한 후 파일로 저장한다.
    ImageIO.write(createImage(Information.getCurrentJPanel()),
        jfilechooser.getFileFilter().getDescription(), x);
} catch (IOException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
break;
}
});

```

❖도형 그리기

- 자유드로잉, 직선/사각형/원드로잉, 지우기, 다각형드로잉, 직선 점 드로잉, 텍스트

```

public void mousePressed(MouseEvent e) { //마우스 프레스 리스너
    //-----위 생략-----
    //나머지 도형 그리기모드
    else { //나머지 도형 그리기모드 일 때(원, 사각형, 자유선 등등)
        drawFigureFunc(e);
    }
}

private void drawFigureFunc(MouseEvent e) {
    int mode = Information.getCurrentMode();
    switch (mode) {
        case Information.MODE_PEN: //펜 모드일 때
            Pen pen = new Pen(e.getX(), e.getY()); //펜 생성
            figureSet.addElement(pen); //배열에 저장
            break;

        case Information.MODE_DRAW_REC: //사각형 모드일 때
            Rectangle rec = new Rectangle(e.getX(), e.getY(), 0, 0); //사각형 생성
            figureSet.addElement(rec); //배열에 저장
            break;

        case Information.MODE_DRAW_CIRCLE: //동그라미 모드일 때
            Circle circle = new Circle(e.getX(), e.getY(), 0, 0); //동그라미 생성
            figureSet.addElement(circle); //배열에 저장
            break;

        case Information.MODE_DRAW_LINE: //선 모드일 때
            Line line = new Line(e.getX(), e.getY(), e.getX(), e.getY()); //선 생성
            figureSet.addElement(line); //배열에 저장
            break;

        case Information.MODE_DRAW_POLYGON: //다각형 모드일 때
            if(count != 10) //10각형 까지 가능
            {
                if(e.getClickCount() == 2) { //더블 클릭 시
                    Polygon polygon = new Polygon(X_point, Y_point, count); //다각형 생성
                    figureSet.addElement(polygon);
                    count = 0; //누른 횟수 초기화
                }
            }
            else {

```

```

        point.add(new Point(e.getX(), e.getY()));
        X_point[count] = e.getX();          //X좌표 저장
        Y_point[count] = e.getY();          //Y좌표 저장
        count++;
    }
}
else {          //10각형
    Polygon polygon = new Polygon(X_point, Y_point, count);
    figureSet.addElement(polygon);
    count = 0;
}
break;
case Information.MODE_ERAGE://지우기 모드일 때
    Eraser eraser = new Eraser(e.getX(), e.getY());//지우개 생성
    figureSet.addElement(eraser);//배열에 저장
    break;
case Information.MODE_TEXT://텍스트 모드일 때
    Text text = new Text(e.getX(), e.getY());//텍스트 생성
    figureSet.addElement(text);//배열에 저장
    RightDownInternalFrame.getInstance().setListItems(figureSet);//RightDownInternalFrame에
                                                //바뀐 정보 추가
    break;
default:
    return;
}
repaint();
}

```

```

public void mouseDragged(MouseEvent e) {//마우스 드래그 리스너
    //-----위 생략-----
    //나머지 그리기
    else {
        Figure current = figureSet.lastElement();//마우스 프레스 리스너에서 추가한 도형 정보를 받아온다.
        figureSet.remove(figureSet.lastElement());//삭제한다.
        drawCurrentFigureFunc(e, current);//
    }
}

public void drawCurrentFigureFunc(MouseEvent e, Figure temp) {
    temp.calcFigure(e.getX(), e.getY());//도형 그리기
    figureSet.addElement(temp);//현재 그리고 있는 도형 정보를 추가한다.
    repaint();
}

public void mouseReleased(MouseEvent e) {//마우스 릴리즈 리스너
    RightDownInternalFrame.getInstance().setListItems(figureSet);//RightDownInternalFrame에 스택정보
                                                //를 업데이트한다.
    figureStack.push((Vector<Figure>) figureSet.clone());//스택에 쌓는다.
    figureCancelStack.clear();//이전 되돌리기 스택을 비운다.
}

```

❖ 색 채우기

- 색상 채우기, 전경/배경 색상선택, 그라데이션 채우기, 투명도

//색칠, 그라데이션

class DrawPanel 에서

```

    public void mouseClicked(MouseEvent e) {//마우스 클릭 리스너

```

```
//-----위 생략-----

//색칠모드
else if (Information.getCurrentMode() == Information.MODE_PAINT) {//현재 모드가 색칠 모드일 때
    if (Information.getCurrentFigure() != null) {//선택된 도형이 있을 때, 그 도형 색 바꾸기
        Information.getCurrentFigure().setFigurePaint(Information.getCurrentColor());
        repaint();
    } else {
        this.setBackground(Information.getCurrentColor());//선택된 도형이 없을 때, 배경 색 바꾸기
        Information.backgroundColor = Information.getCurrentColor();//배경의 정보를 바꾸어준다.
        repaint();
    }
}

//그라데이션 모드
else if(Information.getCurrentMode()==Information.MODE_GRA){//현재 모드가 그라데이션 모드일 때
    if(Information.getCurrentFigure()!=null){//선택된 도형이 있을때
    {
        Information.setGra(Information.getBeforeColor(),Information.getCurrentColor());//이전 색과
        //현재 선택된 색으로 그라데이션을 만든다.
        Information.getCurrentFigure().setFigurePaint(Information.getGra());//도형의 색을 그라데이션
        //으로 바꾼다.

        repaint();
    }
}
}
```

❖객체 선택

- 드로잉된 개체선택기능, 옮기기, 축소/확대

```
public void mousePressed(MouseEvent e) {//마우스 프레스 리스너
    //-----위 생략-----
    //이동모드
    if (Information.getCurrentMode() == Information.MODE_MOVE) {//이동모드 일 때
        if (Information.getCurrentFigure() != null) {//선택된 도형이 있을때
            dragStartX = e.getX();//x좌표를 설정
            dragStartY = e.getY();//y좌표를 설정
        } else {//선택된 도형이 없을 때 경고 오류 메시지를 띄운다.
            JOptionPane.showMessageDialog(null, "Error : Cant' find figure", "ERROR",
JOptionPane.ERROR_MESSAGE);
        }
    }
}

public void mouseDragged(MouseEvent e) {//마우스 드래그 리스너
    //이동모드
    if (Information.getCurrentMode() == Information.MODE_MOVE) {//이동모드일 때
        if (Information.getCurrentFigure() != null) {//선택된 도형이 있으면
            int moveX = e.getX() - dragStartX;이동될 x좌표를 설정한다.
            int moveY = e.getY() - dragStartY;이동될 y좌표를 설정한다.
            Information.getCurrentFigure().moveTo(moveX, moveY);//도형의 좌표를 바꾼다.
            dragStartX = e.getX();
            dragStartY = e.getY();
            repaint();
        }
    }
}
```



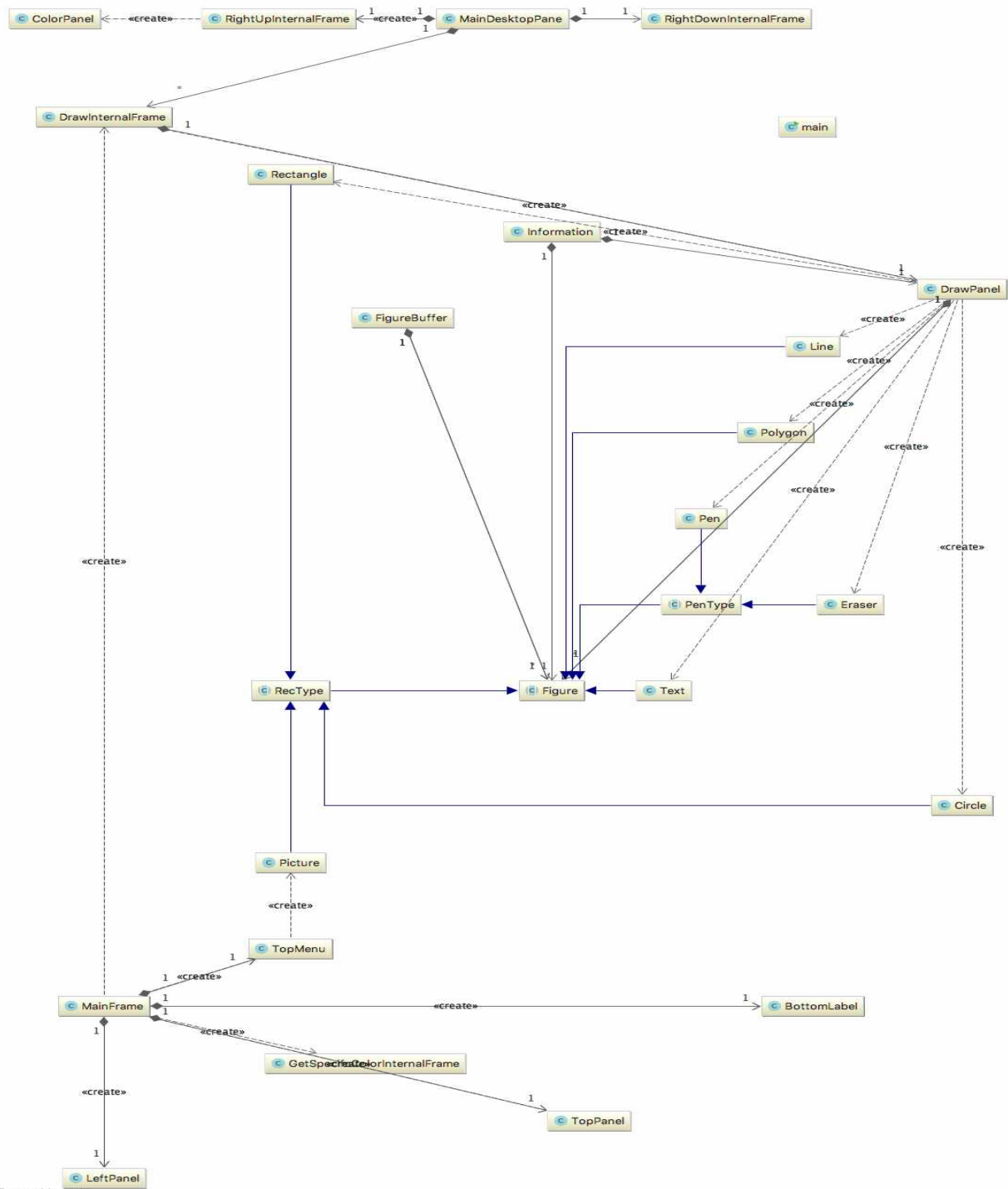
```

//확대 축소 모드
else if (Information.getCurrentMode() == Information.MODE_RESIZE) { //확대축소 모드일 때
    if (Information.getCurrentFigure() != null) { //선택된 도형이 있으면
        Information.getCurrentFigure().calcFigure(e.getX(), e.getY()); //앞에서 그렸던것과 같이
        //다시 그린다.

        repaint();
    }
}
}
}

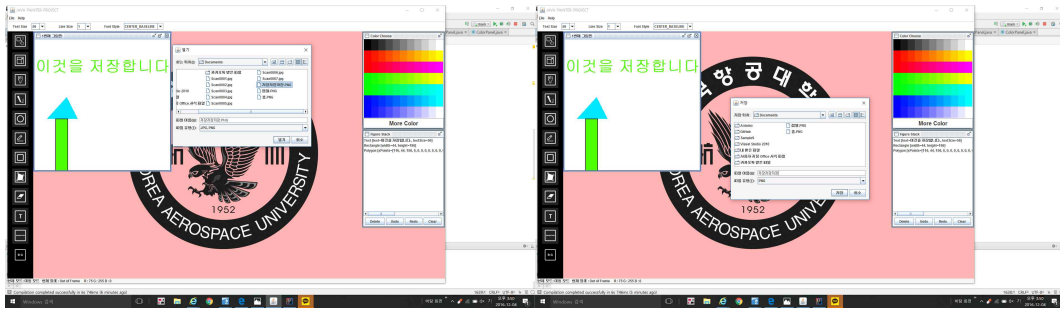
```

❖ Class diagram

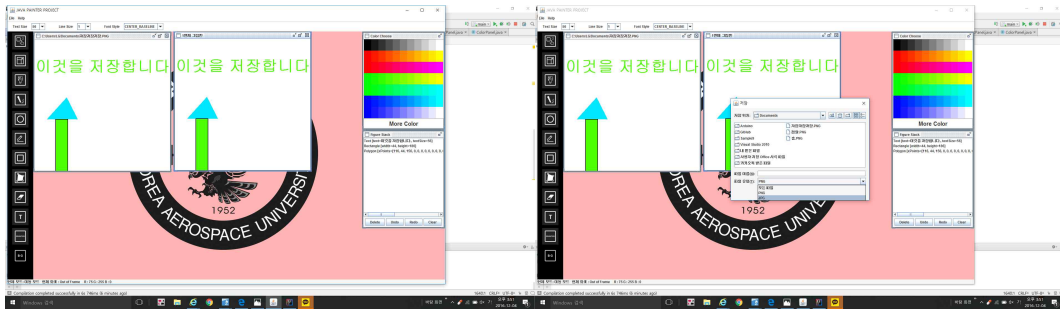


5) 결과

1. 저장/불러오기, JPEG/PNG 등 다른 형식으로 저장/불러오기



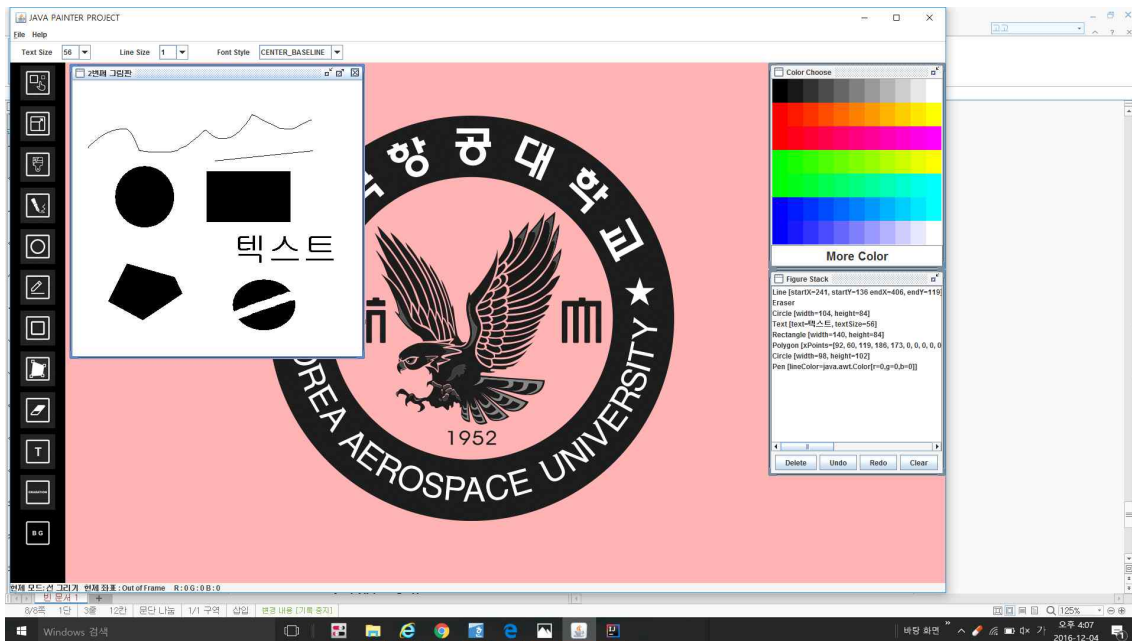
-저장 및 불러오기



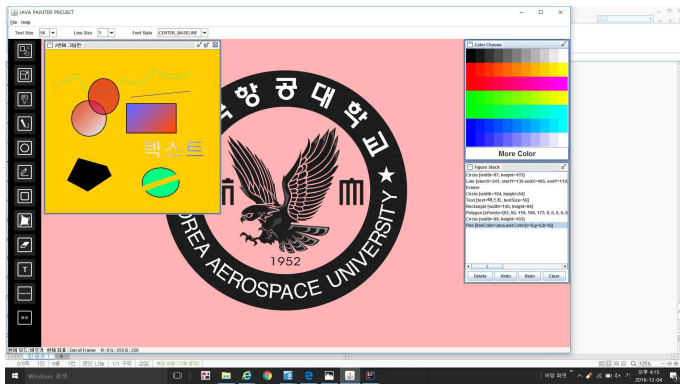
-왼쪽 저장한 파일, 오른쪽 불러온 파일

-jpg, png 선택 가능

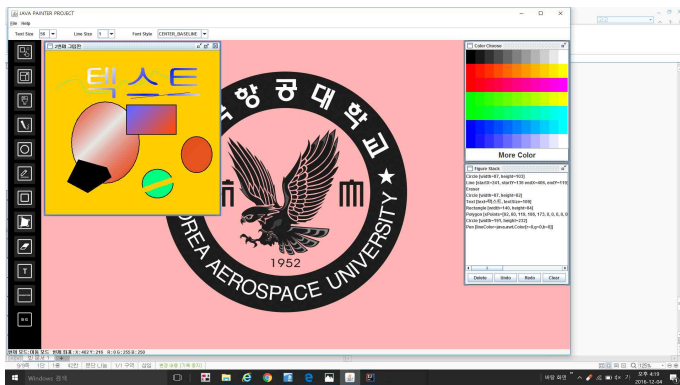
2. 자유드로잉, 직선/사각형/원드로잉, 지우기, 다각형드로잉, 직선 점 드로잉, 텍스트



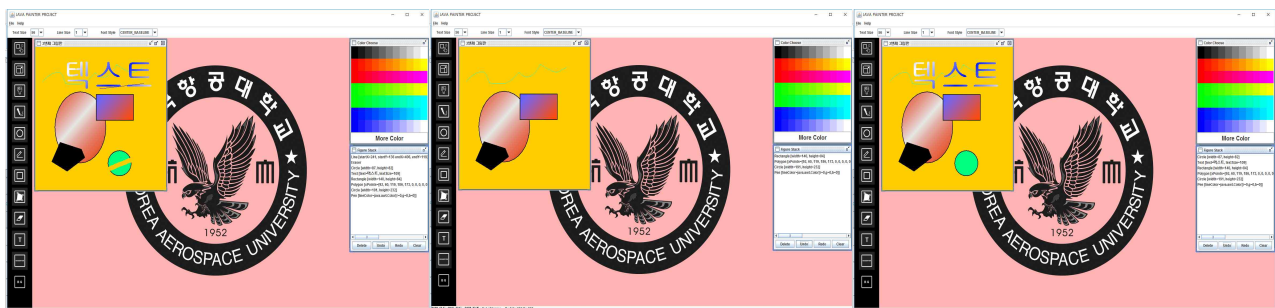
3. 색상 채우기, 전경/배경 색상선택, 그라데이션 채우기, 투명도



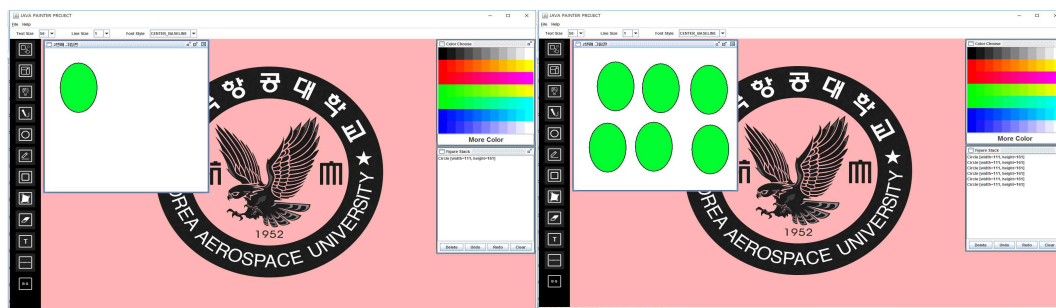
4. 개체선택, 옮기기, 축소/확대



5.히스토리 기능, 복사 붙이기



-히스토리 기능(그리기 -> 되돌리기 -> 되돌리기 취소)



-복사 붙이기 기능 구현

6) 프로젝트 후기

<p>구영민</p>	<p>자바와 객체지향 프로그래밍 과목의 수업을 들으며 자바 언어에 있는 객체 지향 개념을 실제로 어떻게 구현될 수 있는지 궁금했습니다. 메모장 과제를 만들 때에는 객체지향을 제대로 이해하지 못해 C-스타일로 코드를 만들었습니다. 메모장 정도의 작은 어플리케이션의 경우에는 C-스타일로 구현해도 괜찮았지만, 그림판 같은 복잡하고 거대한 프로그램을 구현할 때에는 객체지향 개념을 이용해 클래스 별로 기능을 나누는 게 좋겠다고 회의되어 배운 객체지향 프로그래밍을 최대한 이용하여 코드를 만들었습니다. 저는 파일 저장과 불러오기, 지우개 부분의 구현과 스택 부분을 맡았습니다. 구글을 참고해서 기본 틀을 가져왔는데, 코드가 정말 복잡하여 분석하는데 정말 애를 먹었습니다. 모든 AWT/Swing 객체들은 private으로 래핑되어 있었고, 파일 불러오기를 할 때 불러오는 로직에서 어떻게 하면 그림을 그리는 로직에 추가할 수 있는지에 대해 파악하는 데 시간이 정말 많이 걸렸습니다. 다행히 IntelliJ IDEA 프로그램 덕분에 코드를 분석할 수 있어 작업 시간을 단축할 수 있었습니다. 또 한 가지 자바에서 느꼈던 점은, ImageIO를 이용하면 외부 라이브러리를 사용하지 않고도 일반적인 이미지 파일 입출력을 가능하게 하는 등 유용한 빌트인 라이브러리가 매우 많았다는 것입니다. 자바가 무거운 언어이기는 하지만, 역설적으로 프로그래머가 직접 로우레벨 기능을 구현하지 않아도 이미 많은 부분들이 구현되어 있다는 것을 느꼈습니다. 다행히 JPanel 객체를 어떻게든 가져와 파일 저장과 파일 불러오기 기능을 구현할 수 있었습니다. 그리고 다른 팀원들이 로직 구현에 어려움을 겪을 때 도움을 주었습니다. 일주일동안 2,000줄 정도의 코드를 작성하는 동안 자바 및 객체지향 프로그래밍에 대한 지식을 늘릴 수 있어 정말 뜻깊은 팀프로젝트였습니다.</p>
<p>임희제</p>	<p>이번 그림판 프로젝트를 통해 처음으로 GUI 프로그래밍을 하게 되었는데 그동안 텍스트입력만 가능한 프로그래밍에 비해 직접 화면도 구성하도 다양한 마우스 입력을 받을 수 있어 좀 어려웠지만 새로운 점이 많아 흥미로웠다. 또한 이번 프로젝트를 통해 자바의 객체 지향적 개념을 좀 더 잘 이해하게 된 것 같다. 자바가 객체 지향적 언어다 보니 역할 분담을 편하게 할 수 있었다. 서로 공통된 정보는 Information이라는 class를 통해 공유 하여 변수를 통일하였는데 이렇게 함으로써 코드의 가독성이 쉬워지고 서로의 코드를 합칠 때 마찰이 심하지 않았던 것 같다. 나는 이번에 그라데이션, 투명도, 전경/배경 색상 등의 색에 대한 기능을 주로 맡아 하였는데, Color나 Graphics, Paint등의 자료 형태에 대해 많이 알게 되었다. 특히 그라데이션을 구현 할 때, 그라데이션은 Graphics를 이용해야 되는데 지금까지의 색들이 전부 Color로 되어있어 자료 형태가 맞지 않아 어려움을 겪었었다. 하지만 구글에 계속 검색을 하면서 setPaint라는 메서드를 알게 되었고 setPaint라는 메서드 한 줄을 통해 그동안의 문제점들이 다 해결되어 좀 허무했었다. 지금까지는 그저 알고리즘만을 구현하기 위한 코딩을 하였는데, 이렇게 실생활에 널리 쓰던 그림판을 직접 구현하면서 좀 더 실용적인 공부를 많이 하게 된 것 같다. 일주일 동안 그림판을 만들기에는 매우 촉박한 시간이어서 힘든 점이 많았지만, 구글에 검색을 하면 원했던 기능의 구현 방법을 쉽게 찾을 수 있어 다행이었다. 이번 프로젝트를 통해 구글링 능력과 남의 코드를 해독하는 능력도 많이 향상시킬 수 있었다. 역시 코딩은 하면 되는 것 같다.</p>

지 대 철

‘그림판을 어떻게 일주일안에 만들어?’ 처음 프로젝트를 받게 되었을 때의 생각입니다. 프로젝트가 미리 주어진 것이 아닌 시험 2주전에 발표가 나고 다른 과목의 프로젝트도 겹친 상태라 더욱 조급해 질 수 밖에 없었습니다. 처음회의 때 저희의 목표는 그림판의 기본기능만 구현하는 것이었습니다. 대신 설계과정에서는 기능보다는 객체 지향 프로그래밍과 사용자의 편의에 더 중점을 두자고 의견이 모였습니다. 다행히 2일차에 기본 기능들과 어느 정도의 틀을 만들 수 있었습니다. 우리는 좀 더 욕심을 가졌고 서로 추가기능을 담당하여 구현하기로 했습니다. 제가 맡은 부분은 기본적인 그리기와 응용적인 그리기 등 드로잉과 관련된 부분이었습니다. 코딩부분은 어렵지 않았습니다. 구글에 드로잉에 관한 여러 방법들과 소스들이 있었고 저는 그 소스를 참고하기만 하면 되었습니다. 문제는 각자의 코딩을 하나로 합칠 때 발생했습니다. 객체 지향적으로 설계를 했음에도 불구하고 서로의 코드를 이해하지 못하면 어느 기능에 어떤 클래스를 불러오고 또 어떤 식으로 변경하여야 하는가가 큰 문제였습니다. 이 문제를 해결하기 위해 제가 적용한 방법은 추상클래스입니다. 저는 모든 도형을 추상클래스 없이 각각의 클래스로 설계하여 메서드의 이름도 달랐고 기능 또한 달랐습니다. 도형 부분의 추상클래스 뿐만 아니라 다른 부분에서도 클래스를 연결시키기 위해 다이어그램을 그려보는 등 많은 노력을 통해 코드를 하나로 합칠 수 있었습니다. 그림판의 추가기능을 구현했다는 점도 뿌듯했지만 남에게 미루는 것 없이 밤늦게 까지 모여 같이 코딩하는 조원들한테 고마웠습니다. 추가기능이 구현되자 저희는 인터페이스에 집중했습니다. 다른 조들과는 다르게 독창적인 인터페이스를 설계하고 싶었고 개발자의 입장으로 사용자가 좀 더 편하게 사용할 수 있는 인터페이스를 구현하고 싶었습니다. 비록 대단한 코드나 창의적인 방법이 들어간 것은 아니지만 패널을 추가하고 버튼 이미지 또한 직접 만들고 화면을 여러 개 만들 수 있다는 점에서 다른 조들과 다른 부분을 추가 할 수 있었습니다. 이 점이 프로젝트에 가장 큰 만족도를 주었다고 생각합니다. 피피티 제작과정에서는 모두가 모여 자료 수집, 피피티 작성, 레포트 작성을 맡았습니다. 저의 경우는 필요한 자료를 요청하고 그 자료를 구해주면 피피티를 작성하고 레포트를 작성하는 팀원과 상의하여 수정작업을 걸쳤습니다. 모두가 같이 만든 피피티 인지라 발표의 부담감도 컸습니다. 이번 프로젝트는 저에게 구글에 있는 소스를 참고하는 것이 카피하는 것이 아니라 내 것으로 만드는 과정임을 느끼게 되었고 조원들과의 팀워크가 얼마나 중요한지 알게 되었습니다. 지금까지 했던 프로젝트 중 가장 많은 노력과 좋은 팀워크를 이룬 프로젝트인 만큼 프로젝트의 결과 또한 교수님께서 만족하셨다고 확신합니다.